```php
<?php
/**
 * @author GOLAY Brian
 * @version 1.0 (2021/05/20)
 * Database connection functions
 */
namespace FlightClub\sql;
require_once 'dbInfos.php';

//we check if the user session is already existing. If not we initialize it
if (!isset($_SESSION['user'])) {
    $_SESSION['user'] = "";
}



class DBConnection {
    static $conn = null; //database connection
    const NOM_BASE = DB_NAME;
    const HOST = DB_HOST;
    const USER = DB_USER;
    const PWD = DB_PWD;

    /**
     * Connection to the database function
     *
     * @return object return a non persistant connection object if the db connection was successful, else it return null
     */
    private static function doConnection() {
        try {
            self::$conn = new \PDO(
                    "mysql:host=" . self::HOST .
                    ";dbname=" . self::NOM_BASE, self::USER, self::PWD, array(\PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8",
                \PDO::ATTR_PERSISTENT => false));
            self::$conn->setAttribute(\PDO::ATTR_ERRMODE, \PDO::ERRMODE_EXCEPTION);
        } catch (\Exception $e) {
            echo '<pre>Erreur : ' . $e->getMessage() . '</pre>';
            die('Could not connect to MySQL');
        }
        return self::$conn;
    }

    /**
     * create a new instance of a DB connection if not already existing
     *
     * @return object
     */
    public static function getConnection() {
        if (self::$conn == null) {
            self:: doConnection();
        }
        return self::$conn;
    }

    //TRANSACTION
    /**
     * Begin a transaction
     *
```

```php
     * @return void
     */
    public static function startTransaction()
    {
        DBConnection::getConnection()->beginTransaction();
    }

    /**
     * Rollback a transaction
     *
     * @return void
     */
    public static function rollback()
    {
        try {
            DBConnection::getConnection()->rollback();
        } catch (\Throwable $th) {
            //throw $th;
        }
    }

    /**
     * Commit a transaction
     *
     * @return void
     */
    public static function commit()
    {
        try {
            DBConnection::getConnection()->commit();
        } catch (\Throwable $th) {
            //throw $th;
        }

    }
 }
?>
```